



## 成都远向电子科技有限公司产品说明书

产品型号: YX-AO10V-02

全部资料下载地址: <http://ask.zstel.com:8090>

技术支持服务电话: 028-64267900

技术支持专员企业 QQ: 3183329475

官网网站: <https://www.cdyxiot.com/>

硬件/软件技术定制热线: 19150158475 张工

# 目录

目录.....	2
一、 产品概述.....	4
1.1 概述.....	4
1.2 性能特性.....	4
1.3 技术参数.....	5
二、 外观尺寸.....	6
2.1 产品外观.....	6
2.2 产品尺寸图.....	6
三、 产品接线图、跳线、指示灯说明.....	8
3.1 端子接口.....	8
3.2 接线图.....	8
3.3 跳线.....	错误！未定义书签。
3.4 LED 指示灯.....	9
四、 ModbusRTU 通讯协议地址以及案例说明.....	10
4.1 通讯协议.....	10
4.2 寄存器地址.....	10
4.3 Modbus RTU 功能码.....	10
4.4 Modbus 通讯实例.....	11
五、 软件操作.....	13
5.1 配置软件.....	13
5.2 配置基本参数.....	14

5.3 模拟量通道校准相关参数 .....	15
5.4 其他功能 .....	17
六、协议详解 .....	错误！未定义书签。
6.1 功能码描述 .....	错误！未定义书签。
6.2 CRC 校验算法 .....	21
七、更改记录 .....	25

## 一、产品概述

### 1.1 概述

YX-AO10V-02 是一款工业级标准模拟量采集-输出产品，共有 模拟量 2 个测量通道。每个通道均可以分别设置多种量程。拥有 2 个模拟量输出通道，可实现 0.5%精度模拟量输出数值化控制。RS-485 通讯接口使用标准 Modbus RTU 协议，符合工业标准。

### 1.2 性能特性

- 防死机硬件看门狗
- 12 ~ 36V 带防反接、过压过流保护电源
- 2 路模拟量电压输入 0 ~ 30V
- 2 路模拟量电压输出 0 ~ 10V
- 12 位分辨率，AI 输入 0.1%精度，AO 输出 0.5%精度
- 高性能低功耗 32 位 ARM 嵌入式 CPU
- 支持 ModbusRTU 从站协议
- 1路指示灯
- 带防雷、静电保护 RS485 通讯接口
- 工业机温度范围，应对严苛现场环境
- 自定义线性模拟量数据转换
- 支持标准导轨安装

### 1.3 技术参数

模拟量输入接口	AI	2 路单端
	AI 分辨率	12bit
	AI 量程	0 ~ 30V
	精度	0.1%
	采集速度	单通道固定 1kHz
	AI 输入阻抗	0 ~ 30V ≥ 10KΩ
模拟量输出接口	AO	2 路单端
	AI 分辨率	12bit
	AO 量程	0 ~ 10V
	精度	0.5%
	负载阻抗建议	100 ~ 250Ω
通讯接口	通讯接口	RS485
	波特率	1200 ~ 115200bps
	数据格式	8N1,8O1,8E1,8N2,8O2,8E2
	通讯协议	ModbusRTU
	过压保护	45V
电源参数	电源规格	DC 12 ~ 36V
	功耗	12V-0.5W
工作环境	工作温度、湿度	-40℃ ~ 85℃, 0%RH ~ 95%RH
其他	尺寸	82*54*32

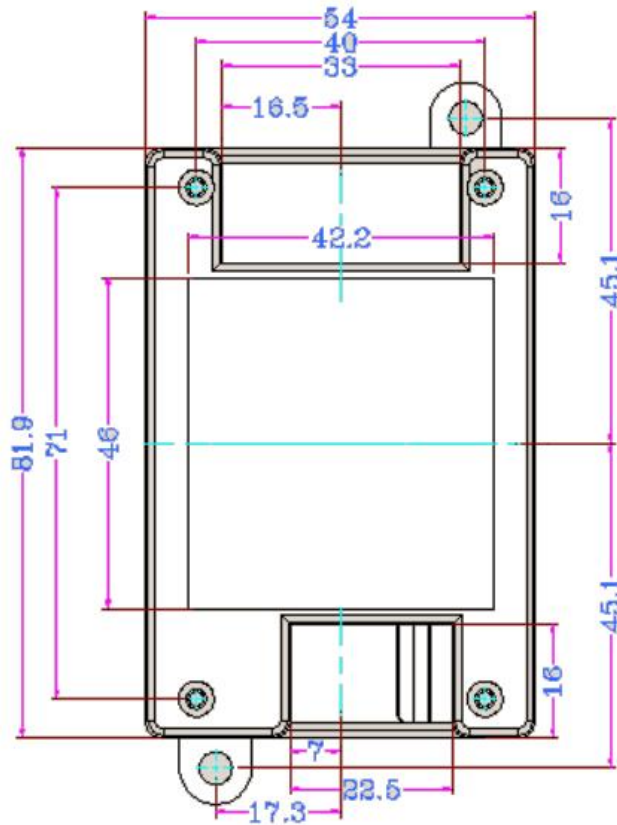
## 二、外观尺寸

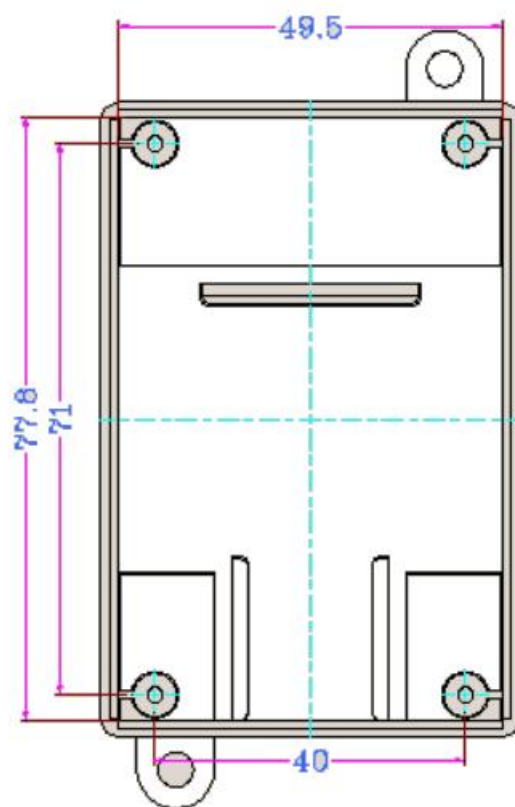
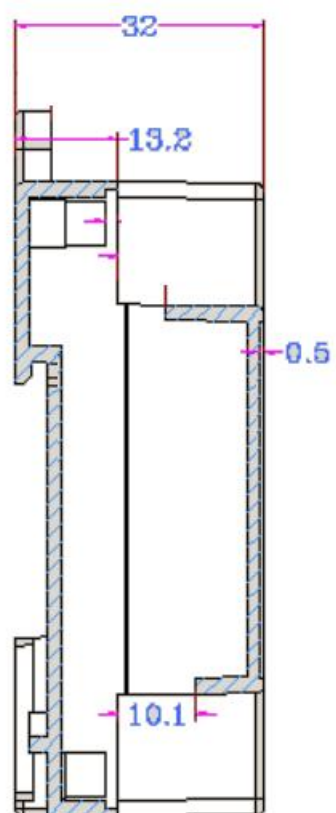
### 2.1 产品外观



长：82，宽：54，高：32MM

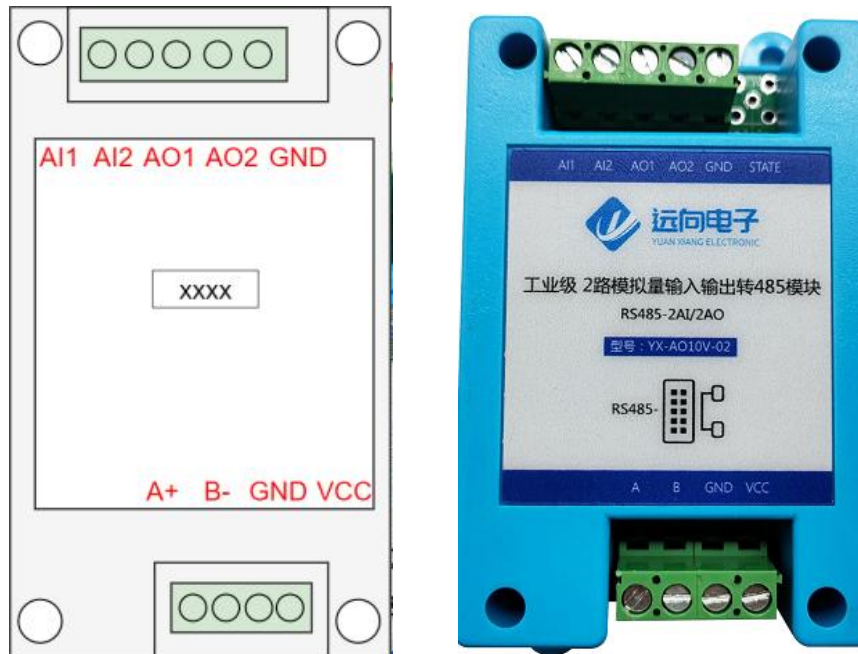
### 2.2 产品尺寸图





### 三、产品接线图、指示灯说明

#### 3.1 端子接口



- 顶部 5 槽接线位:

AI1: 模拟量输入通道 1  
AI2: 模拟量输入通道 2  
AO1: 模拟量输出 AO1  
AO2: 模拟量输出 AO2  
GND: 电源负极

- 底部 4 槽接线位:

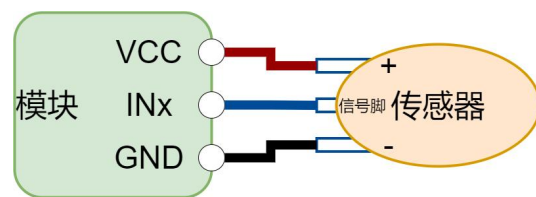
A+: RS485 通讯线 A  
B+: RS485 通讯线 B  
GND: 电源负极  
VIN: 电源正极

#### 3.2 接线图

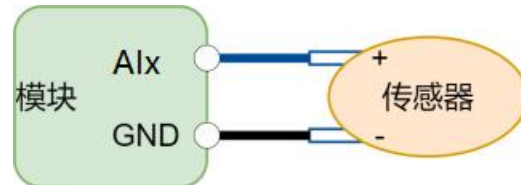
(1) 模拟量输入接线图



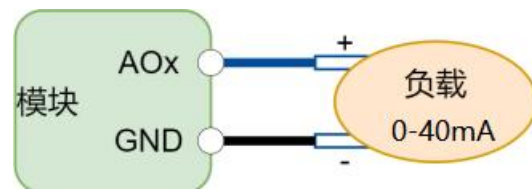
有源传感器（3 线制）：



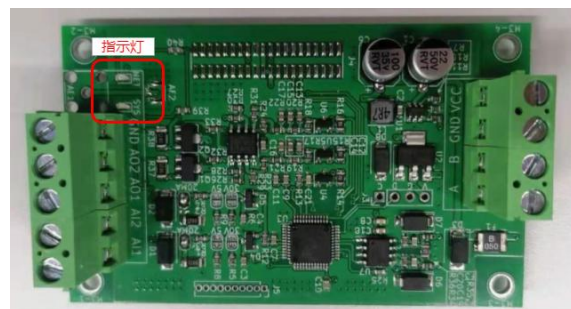
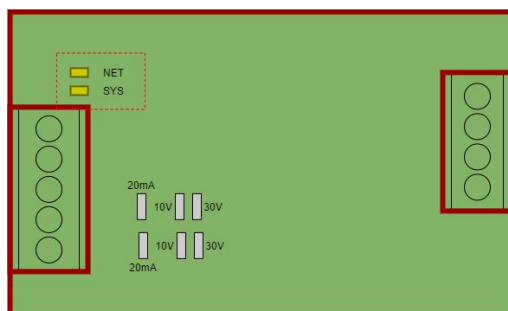
有源传感器(2 线制)：



(2) 模拟量输出接线图



### 3.4 LED 指示灯



2 个 LED 指示灯：

- **SYS**：系统状态灯，正常运行时每秒闪烁一次，有串口数据通信时也会闪烁

## 四、ModbusRTU 通讯协议地址以及案例说明

### 4.1 通讯协议

本产品支持标准 Modbus RTU 从站协议，能够支持标准 Modbus RTU 组态软件，详细内容介绍参考本文第六章内容

### 4.2 寄存器地址

寄存器地址	名称	字节数	说明	备注
模拟量输入				
0x0000(0)	AI1_H	2	模拟量输入通道 1 高	每个模拟量通道占 2 个 Modbus 寄存器，4 个字节，格式为浮点数，浮点数格式符合 IEEE 754 标准 0~30V: 0~30.0
0x0001(1)	AI1_L	2	模拟量输入通道 1 低	
0x0002(2)	AI2_H	2	模拟量输入通道 2 高	
0x0003(3)	AI2_L	2	模拟量输入通道 2 低	
0x0100(256)	AI1_D	2	模拟量输入通道 1	AI 整数寄存器 0~30V: 000~3000
0x0101(257)	AI2_D	2	模拟量输入通道 2	
模拟量输出				
0x1100(4352)	AO1_D	2	模拟量输出通道 1	AO 整数寄存器 0~10V: 0~4095
0x1101(4353)	AO2_D	2	模拟量输出通道 2	

### 4.3 Modbus RTU 功能码

功能码	操作	说明
01	读取单位 DO 状态	Bit 位表示 DO 输出状态
03	读取 AI, DO 寄存器值	读取 AI, DO 寄存器值
04	读取 AI, DO 寄存器值	读取 AI, DO 寄存器值
05	写单个 DO	0xFF00: 闭合;0x0000: 断开

06	写单个 DO	0x0001: 闭合;0x0000: 断开
0F	写多个 DO	参照本文第六章内容
10	写多个 DO	参照本文第六章内容

详细讲解参照本文第六章内容

## 4.4 Modbus 通讯实例

### (1) 读取 AI:

给定输入 4.96

#### a. 用 03 功能码读取浮点数 AI1:

发送: 01 03 00 00 00 02 C4 0B

接受: 01 03 04 40 9E E7 CF 85 B9

4.96 IEE 浮点数十六进制为 (40 9E E7 CF)

#### b. 用 04 功能码读取浮点数 AI1:

发送: 01 04 00 00 00 02 71 CB

接受: 01 04 04 40 9E CE 1F 9A 02

#### c. 用 03 功能码读取整数 AI1:

发送: 01 03 01 00 00 01 85 F6

接收: 01 03 02 01 F0 B9 90

整数读出数值为 496 (0x01F0)

#### d. 用 04 功能码读取整数 AI1:

发送: 01 04 01 00 00 01 30 36

接收: 01 04 02 01 F0 B8 E4

整数读出数值为 496 (0x01F0)

## (2) 设置 AO 输出:

设置 AO 输出 4.9V 模拟量

### a. 用 10 功能码写浮点数 AO

发送: 01 10 10 00 00 02 04 40 9C CC CD 7F 14

接受: 01 10 10 00 00 02 45 08

测量输出端即可量出 2.495V 电压

### b. 用 10 功能码写整数 AO

发送: 01 10 11 00 00 01 02 03 EB E7 EE

接受: 01 10 11 00 00 01 04 F5

设置整数寄存器值为 1003 ( $1003 = 4095 * 2.495 / 10$ ) 测量输出端即可量出

2.495V 电压

### c. 用 03 功能码读取浮点数 AO 值

发送: 01 03 10 00 00 02 C0 CB

接受: 01 03 04 40 9C CC CD BB 48

### d. 用 03 功能码读取整数 AO 值

发送: 01 03 11 00 00 01 81 36

接受: 01 03 02 03 EB F8 FB

## 五、软件操作

设备参数配置教程，结合《用户测试文档》即可对设备进行简单测试

### 5.1 配置软件

参数配置软件介绍：



5.1.1 配置软件包含有:

- **功能区:** 包含有配置软件所支持功能,以及功能那个切换选项
- **参数配置主区域:** 参数配置主要区域, 参数项的读取、写入临时列表
- **串口/命令集区:** 涉及模块的参数读、写、重启等操作
- **串口日志区:** 命令集的操作日志

5.1.2 参数配置准备:

- (1) 用 USB-485 工具连接设备到电脑
- (2) 在串口配置框内配置串口波特率、停止位、校验位、数据位; (默认波

特率 9600，数据位 8，停止位 1，校验位 None)

(3) 选择串口配置框子项“命令集”



(4) 点击“读取参数”命令按钮，读取设备参数（不同设备拥有不同指令集）

(5) 双击对应参数项的“参数值”，然后对参数进行修改

(6) 修改完参数后需要点击命令集里的“设置参数”，写入到模块中

(7) 写入完成在日志区域会提示成功。



(8) 通过点击“重启设备”按钮，重启模块设备使配置参数生效

5.2 配置基本参数

该系列参数涉及到对 485 通讯 Modbus 协议相关配置。

参数名称	参数值	参数说明
<基本参数>		
Modbus地址	双击修改参数	设备的Modbus地址, 1~255
通信模块波特率		与通信模块的波特率一致, 一般设置为9600
通信模块奇偶校验		与通信模块的奇偶校验一致, 一般设置为8N1

- Modbus 地址: Modbus 地址参数,可设置 1 ~ 255

- 通讯模块波特率：设备 485 通讯波特率（波特率支持主流的波特率选项）
- 通讯模块就校验：设备 485 通讯奇偶校验位，可配置 8N1,8E1,8O1...

### 5.3 模拟量通道校准相关参数

本系列参数用于模拟量读数转换，RTU 采集模块默认读取出来的读数值为采集的电压值，接传感器时有时会用到将电压值转换为实际传感器读数。则可以使用本系列参数实现读数转换功能。

<AI1模拟通道校准相关参数>		
AI1增益值 gain		AI1采集的模拟量计算公式 AI = ( adc + gain ) * ratio + offset
AI1比例值 ratio	双击修改参数	AI1采集的模拟量计算公式 AI = ( adc + gain ) * ratio + offset
AI1位移值 offset		AI1采集的模拟量计算公式 AI = ( adc + gain ) * ratio + offset

- AIx 增益值 gain: AIx 数值转换公式中的增益值(详细使用参考后续说明)
- AIx 比例值 ratio: AIx 数值转换公式中的比例值(详细使用参考后续说明)
- AIx 偏移值 offset: AIx 数值转换公式中的偏移值(详细使用参考后续说明)

#### 5.4.1 转换原理讲解

模拟量读数转换公式如下（例：将电压转换为温度值）

$$AI = (adc) \times ratio + offset$$

AIx 即转换后数值：

- adc 为转换前读数值
- ratio 为设定比例系数（初始为 1.0）
- offset 为设定位置系数（初始为 0.0）

$$ratio = \frac{\text{传感器}_{max} - \text{传感器}_{min}}{\text{模拟量}_{max} - \text{模拟量}_{min}}$$

$$offset = \frac{\text{模拟量}_{max} * \text{传感器}_{min} - \text{模拟量}_{min} * \text{传感器}_{max}}{\text{模拟量}_{max} - \text{模拟量}_{min}}$$

### 5.4.2 校准实例

本案例举例说明如何将一个 0 ~ 10V 对应 -40℃ ~ 120℃ 的温度传感器输出的电压值转换为实际温度读数。

- ◆ 已知温度传感器 输出模拟量范围是 0 ~ 10V, 将 RTU 采集模块量程设置为 0-30V 量程范围。
- ◆ 计算需要比例 (ratio)、偏移 (offset) 三个参数值。

根据传感器参数可知

$$\text{模拟量}_{min} = 0$$

$$\text{模拟量}_{max} = 10$$

$$\text{传感器}_{min} = -40$$

$$\text{传感器}_{max} = 120$$

代入 ratio、offset 计算公式中可得

$$ratio = [120 - (-40)] / (10 - 0) = 16$$

$$offset = \{[10 * (-40)] - [0 * (120)]\} / (10 - 0) = -40$$

- ◆ 根据将计算后的参数代入转换公式即可得

传感器输出模拟量 4V ---> 传感器温度读数 = 4\*16-40 = -24℃



传感器输出模拟量 10V --> 传感器温度读数 =  $10 \times 16 - 40 = 120^{\circ}\text{C}$

## 5.5 其他功能

### 5.5.1 校准模拟量精度

当系统精度不够时，产品提供一个用户精度校准功能

(1) 设备串口连接上电脑，点击“命令集”中“校准模拟量精度”按钮进行模拟量校准



(2) 取消勾选显示区“HEX”复选框，等到接受区提示如下内容时执行下一步

数据收发区

已接收:4819字节  已发送:189字节

```

【14:20:24.693】 下行:CH1: 10
【成功】
【14:20:31.317】 上行:■U
【14:20:36.473】 上行:
校准系数: 1.0000, 1.0002
串口等待接受校准模拟量数值
串口发送格式如下:
CH1: xx
CH2: xx
【14:20:38.078】 下行:CH1: 10
【14:20:38.238】 上行:模拟通道1校准完成
【14:20:38.400】 上行:模拟通道1系数: 0.9986
【14:20:38.597】 上行:退出模拟量校准模式
【重启设备】
【14:20:45.546】 下行:AA550004E00600EA
【成功】
【14:20:45.719】 上行:■U
【成功】
【14:20:46.325】 上行:HIGH

```

☐ HEX ☒ 显示时间 ☒ 自动换行

CH1: 10

☐ HEX ☒ 显示发送 ☐ 自动发送

(3) 给定模拟量 CHx 通道恒定基准信号（注意通道量程）

(4) 发送通道基准信号格式 “CHx: xx”（例：“CH1: 10”）注意取消勾选发送区

“HEX”复选按钮

(5) 等待设备提示校准消息

数据收发区

已接收:4819字节

收清零

已发送:189字节

发清零

【14:20:24.693】 下行:CH1: 10

【成功】

【14:20:31.317】 上行:■U

【14:20:36.473】 上行:

校准系数: 1.0000, 1.0002

串口等待接受校准模拟量数值

串口发送格式如下:

CH1: xx

CH2: xx

【14:20:38.078】 下行:CH1: 10

【14:20:38.238】 上行:模拟通道1校准完成

【14:20:38.400】 上行:模拟通道1系数: 0.9986

【14:20:38.597】 上行:退出模拟量校准模式

【重启设备】

【14:20:45.546】 下行:AA550004E00600EA

【成功】

【14:20:45.719】 上行:■U

【成功】

【14:20:46.325】 上行:HIGH

☐ HEX
☒ 显示时间
☒ 自动换行

清空接受区

CH1: 10

☐ HEX
☒ 显示发送
☐ 自动发送

清空发送区

发送

(6) 重启设备

## 5.5.2 RTU 数据可视化测试

参数配置软件包含有 RTU 数据可视化功能,可实现简单的模块功能测试。

操作步骤如下图：



- 选择"RTU 可视化"界面
- 配置 USB 转 485 通讯参数并打开串口
- 设置 Modbus 子站地址为默认"1"
- 點選"自动读取"复选框

注意: 此时界面左侧仪表盘会根据模拟量信号源的输出显示对应电压读数, 右侧会有上下行通讯报文。若无上行数据请检查 USB-485 转换器是否正常工作, 或尝试模块 485 AB 之间接 120 欧电阻。

详细的测试演示可参考["用户测试文档"](#)

## 6.1 错误码描述

错误码含义：当 DTU 收到错误的 Modbus 指令时，会返回功能码为请求功能码+0x80，紧随着一个字节的错误码代表出错原因。

错误码 01：表示不支持的功能码，RTU 支持上述 8 种功能码，除此之外的功能码都会返回错误码为 01 的错误。

错误码 02：表示起始地址不存在或者起始地址加上寄存器数量后的地址不存在。总的来说表示访问的寄存器不存在。

错误码 03：表示寄存器数量不符合规范或者寄存器值非法。

错误码 04：表示读写寄存器错误。

## 6.2 CRC 校验算法

CRC 即[循环冗余校验码](#)（Cyclic Redundancy Check）：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查（CRC）是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

ModbusRTU 的 CRC16 计算初值: 0xFFFF

ModbusRTU 的 CRC16 计算多项式 0xA001 (二进制:1010 0000 0000 0001)

附 CRC 校验算法代码：

```
uint16_t mb_crc( uint8_t* snd, uint16_t num )
```

```
{
```

```
    uint8_t CRC_Lb, CRC_Hb;
```

```
    uint8_t CRC_L, CRC_H;
```

```
    uint16_t crc;
```

```
    CRC_H = 0xFF;
```

```
    CRC_L = 0xFF;
```

```
    for ( uint16_t i = 0; i < num; i++ ) {
```

```
        CRC_L = CRC_L ^ snd[ i ];
```

```
        for ( uint16_t j = 0; j < 8; j++ ) {
```

```
            CRC_Lb = CRC_L;
```

```
            if ( ( CRC_L & 1 ) == 1 ) {
```

```
                CRC_L = ( CRC_L - 1 ) / 2;
```

```
                CRC_Lb = CRC_L;
```

```
                CRC_Hb = CRC_H;
```

```
            if ( ( CRC_H & 1 ) == 1 ) {
```

```
                CRC_L = CRC_L + 128;
```

```
                CRC_Lb = CRC_L;
```

```
                CRC_H = ( CRC_H - 1 ) / 2;
```

```
                CRC_Hb = CRC_H;
```

```

    } else {

        CRC_H = CRC_H / 2;

        CRC_Hb = CRC_H;

    }

    CRC_L = CRC_L ^ 1;

    CRC_Lb = CRC_L;

    CRC_H = CRC_H ^ 0xA0;

    CRC_Hb = CRC_H;

} else {

    CRC_L = CRC_L / 2;

    CRC_Lb = CRC_L;

    CRC_Hb = CRC_H;

    if ( ( CRC_H & 1 ) == 1 ) {

        CRC_L = CRC_L + 128;

        CRC_Lb = CRC_L;

        CRC_H = ( CRC_H - 1 ) / 2;

        CRC_Hb = CRC_H;

    } else {

        CRC_H = CRC_H / 2;

        CRC_Hb = CRC_H;

    }

}

```

```
    }  
  
}  
  
crc  = CRC_L;  
  
crc <<= 8;  
  
crc |= CRC_H;  
  
return crc;  
}
```



## 七、更改记录

v1.6

- \* 新增 AI 整数段寄存器读取说明以及示例

v1.5

- \* 更新有源、无源接线方式

v1.4

- \* 更新外壳尺寸图
- \* 更新电池检测接线方式

v1.3

- \* 接线图新增三线制

v1.2

- \* 修改接线图，调整文字字体

v1.1

- \* 修改 3.1 接线图，3.2 跳线图，3.3LED 指示灯配图

v1.0

- \* 第一版编写

